scientific reports

Check for updates

OPEN

A reinforcement learning approach for reducing traffic congestion using deep Q learning

S M Masfequier Rahman Swapno[1], SM Nuruzzaman Nobel[1], Preeti Meena[2], V. P. Meena[3]✉, Ahmad Taher Azar[4,5,6]✉, Zeeshan Haider[4,5] & Mohamed Tounsi[4,5]✉

Nowadays, traffic congestion is a significant issue globally. The vehicle quantity has grown dramatically, while road and transportation infrastructure capacities have yet to expand proportionally to handle the additional traffic effectively. Road congestion and traffic-related pollution have increased, which is detrimental to society and public health. This paper proposes a novel reinforcement learning (RL)-based method to reduce traffic congestion. We have developed a sophisticated Deep Q-Network (DQN) and integrated it smoothly into our system. In this study, Our implemented DQL model reduced queue lengths by 49% and increased incentives for each lane by 9%. The results emphasize the effectiveness of our method in setting strong traffic reduction standards. This study shows that RL has excellent potential to improve both transport efficiency and sustainability in metropolitan areas. Moreover, utilizing RL can significantly improve the standards for reducing traffic and easing urban traffic congestion.

Keywords Traffic reduction, RL, Smart city, DQL, Queue length, Rewards, Intersection, Agent

Abbreviations

DQL	Deep Q learning
RL	Reinforcement learning
ITS	Intelligent transportation systems
ICT	Information communication technology
IOT	Internet of things
TMS	Traffic management system
TSC	Traffic signal control
MARL	Multi-agent reinforcement learning
SARL	Safety aware reinforcement learning
TC	Traffic control
DP	Dynamic programming
ID	Ddentification
NSA	North south arms
NSLA	North south left arms
EWA	East west arms
EWLA	East-west left arms
DQN	Deep Q Network
TSP	Transit signal priority
ATSC	Adaptive traffic signal control

Urban areas[1,2] are more and more facing the problem of traffic congestion. In the transportation sector, this issue dramatically impacts travel time, fuel consumption, an operating costs. Moreover, congestion significantly contributes to pollution, resulting in a severe environmental impact[3,4]. Several studies have been conducted to

[1]Department of CSE, Bangladesh University of Business and Technology, Dhaka, Bangladesh. [2]Department of Electrical Engineering, Indian Institute of Technology, Jodhpur, Rajasthan 342030, India. [3]Department of Electrical Engineering, National Institute of Technology Jamshedpur, Jamshedpur 831014, Jharkhand, India. [4]College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia. [5]Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. [6]Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt. ✉email: vmeena1@ee.iitr.ac.in; aazar@psu.edu.sa; mtounsi@psu.edu.sa

develop efficient traffic management systems to address this pressing issue. Recent initiatives have concentrated explicitly on Intelligent Transportation Systems (ITS). These efforts aim to improve the safety, effectiveness, and environmental sustainability of traffic control systems. Researchers aim to develop creative solutions within the ITS field to reduce congestion and enhance urban sustainability[5]. It is essential, to address critical aspects[6] such as lowering delay and queue length to optimize traffic flow at an intersection. Traffic congestion can be reduced by improving traffic management that is closely linked to the intersection routes' layout and structure. Implementing strategic initiatives to optimize and improve traffic flow on these routes is crucial for attaining a smoother and more efficient traffic experience[7,8]. By efficiently reducing congestion in several routes at an intersection, They automatically decrease the overall traffic flow. By strategically managing and reducing congestion on particular routes[9], we have created a smoother and more synchronized traffic movement. This method improves the intersection's efficiency and helps creating a smoother, less crowded transit network.

Motivation and contribution of this research

The motivation behind this research stems from the pressing global issue of traffic congestion, which has become a significant challenge due to the rapid increase in vehicle numbers without corresponding expansion in transportation infrastructure. This imbalance has led to numerous adverse consequences, including increased road congestion and pollution, which have far-reaching impacts on society and public health. To address this critical problem, we aim to leverage advanced technologies, specifically RL, to devise innovative solutions for reducing traffic congestion. RL, particularly DQN, presents a promising approach that harnesses computational intelligence to optimize traffic flow and alleviate congestion effectively. By developing and implementing a sophisticated DQL model within a transportation system, the researchers seek to demonstrate the transformative potential of RL in enhancing transport efficiency and sustainability in urban areas. The study's motivation is grounded in the urgent need to adopt intelligent, data-driven approaches to tackle traffic congestion, promoting safer, more efficient, and environmentally sustainable urban mobility. The expected outcomes of this research include setting new traffic reduction standards and advancing urban transportation systems by applying cutting-edge RL methodologies. Ultimately, this work aims to contribute to developing more intelligent, more adaptive urban transportation networks capable of addressing the challenges posed by growing vehicle populations and limited infrastructure capacities.

This study presents the development of an advanced traffic reduction system that utilizes intelligent technologies to minimize delays. We proposed a RL framework for the system, a type of machine learning, to efficiently optimize traffic flow and reduce congestion. Our intelligent traffic management technology combines sophisticated algorithms with up-to-date data to minimize delays and improve overall efficiency. This novel approach signifies substantial progress in traffic control systems, possibly resolving current urban mobility difficulties. The contribution of this research is as follows:

- Focusing on specific benchmark methods to ensure successful traffic reduction implementation for enhancing a traffic-free smart city.
- Applying advanced, customized layer based method for making efficient traffic reduction.
- Developing an advanced DQN to maintain the traffic reduction system in an intersection.
- Performing RL technique of state, action, and rewards successfully in the traffic reduction domain.
- Focusing on minimizing queue length and increasing rewards at each length.

Organization of the paper

The research is divided into multiple areas, each with a specific function. Section "Related work" summarizes relevant literature on the topic and serves as the basis for the investigation. Section "Research methodology" details the research design and methodology employed in the study. The study's findings are detailed in Section "Result analysis", while. Section "Discussion" contains a thorough topic analysis and a critical assessment of the results and their consequences. Section "Conclusion and future work" concludes our work and provides suggestions for future research directions.

Related work

Machine learning and deep learning are important parts of many fields these days, like healthcare, banking systems, and business management[10–27]. Our current study focuses on machine learning applications for improving traffic signal management in urban areas. The main goal is to create a functional congestion system and reduce waiting times and line lengths. Due to this endeavor, many researchers have used various methods to reduce traffic-congestion. Interestingly, the scene displays many innovative methods currently being used to control traffic signals in cities.

Various RL schemes have been combined with centrally performed genetic algorithms for parameter tweaking to produce "intelligent" cooperation schemes among RL-based traffic control agents, as demonstrated in[28]. A type of RL in conjunction with fuzzy neural networks is used to construct the hierarchical real-time traffic control architecture as shown in[29]. The authors[30] examined the advantages of multi-agent model-based RL for traffic control from a vehicle-centric perspective. A similar method is provided in[31]. These two approaches make some significant assumptions about the required information and may not be realistically obtained, particularly when traffic patterns change and drivers' personalities are considered. Examples of this information include probability estimates of waiting times for each vehicle's destination and their location at each traffic light controller, including whether the light is green or red.

The multi-agent reinforcement learning (MARL) technique is an alternative signal control method involving numerous signalized intersections. Each junction is under the control of a different RL agent, similar to safety-aware reinforcement learning (SARL). Large networks could benefit from the adoption of MARL techniques.

Three algorithms, TC1, TC2, and TC3, are presented by the authors of[30], who proposed a method for adaptive signal regulation based on RL. The results demonstrate that RL performed better on an essential 3:2 grid when using fixed time controllers. Additionally, the authors use co-learning to teach signal controllers and driving agents value functions. Learning to compute policies can also be used to find the network's optimal paths. This was the most fundamental task, with numerous restrictions, including fixed time controllers and state space increments. By implementing essential information sharing between RL agents,[32] expands on this work. The authors presented three new traffic controllers-TC-SBC, TC-GAC, and TCSBC+GAC. To determine the traffic conditions at nearby intersections, TC-SBC adds a small quantity of congestion. TC-GAC selects the best course of action based on this information. Tested algorithms on essential grid networks fared well in varying traffic scenarios. The drawbacks of TC-SBC include its difficulty in computing growth in state space size and the fact that TC-GAC is not perpetually aware of traffic patterns or congestion. In[33], two new algorithms, TC-SBA and TCSBAC, are introduced to improve this work further. The vast state space again causes it to suffer from adding extra bits to the state and mishaps-four times as much as the initial TC1 is exceeded in the new state space. In[34], MARL is also implemented on big networks with almost 50 junctions. Based on Q-learning, the technique considers the average queue length at each link on every junction for state representations. Action selection is based on a green time ratio and fixed time signal designs.

Partial detection is currently being used in a few research projects. One initiative that focuses on low-penetration rates of automobiles equipped with dedicated short-range communications (DSRC) is called COLOMBO, for instance[35,36]. The system feeds data to a traffic control system using information obtained from vehicle-to-everything (V2X) technology. Under low to medium car flow, COLOMBO will only achieve optimal performance if the optimal strategy under low-to-medium car flow must react by detected car arrivals. This is because COLOMBO cannot directly react to real-time traffic flow, as detected and undetected vehicles perform similarly. DSRC actuated traffic lights are another relatively new method they have used to control traffic utilizing DSRC radio. In July 2018, a public demonstration of the system's designed prototype took place in Riyadh, Saudi Arabia[37–39]. On the other hand, because DSRC actuated traffic lights depend on every vehicle's arrival, they function best in low- to medium-car flow rates and poorly in high-car flow rates.

The authors address the safety and efficiency concerns of the stage-based signal control strategy[40] in their group-based optimization model for mixed traffic flows with uneven volumes. A unique genetic algorithm-based method for optimizing transit priority in situations involving both private and transit traffic is proposed in[41]. A simulation-based generic algorithm with a multi-objective optimization model that considers emissions and delays is used[42]. Simulation-based techniques are also frequently used for the TSC problem. The authors of[43] have put out a strategy that uses local communication between the sensors and the traffic lights to manage them in a way that responds to traffic in intricate real-world networks. Referencing[44], an analysis of the CRONOS algorithm's behavior revealed that it reduces overall delay compared to local and centralized control strategies. Because dynamic programming (DP) is very adjustable, can be employed under a wide range of traffic conditions, and can leverage a wide range of performance indicators, it has been utilized by many researchers to solve the TSC problem. By approximating the value function, the authors of[45] employed an approximate dynamic programming-based method that significantly decreased vehicle delays while reducing the processing resources needed. RHODES breaks down and restructures the TSC problem into hierarchical sub-problems, as shown in[46]. The study also demonstrated that RHODES works better than semi-actuated controllers in terms of delay.

We have recognized significant advancements in the field of traffic signal control involving the utilization of methodologies and algorithms. Multiple academics, such as Tan et al.[47], Wan et al.[48], Gender et al.[49], and Gao et al.[50], have studied rewards and state functions, using neural schema to improve traffic light control. They are known for their groundbreaking efforts to improve efficiency using new methods. Similarly, academics like Mousavi et al.[51], Liang[52], and Van et al.[53] have focused on TSC. They focus on improving the TSC system by reducing delays and queue lengths. Their findings demonstrate significant excellence. After a thorough comparison with current methods, our system proves to be notably more efficient. This comparison emphasizes the development in traffic reduction and showcases the historical importance of certain specific techniques. However, our system has outperformed existing research in terms of outcomes. Additionally, our system is highly dependable and has obtained the most efficient results. Table 1 summarizes a neural network-based comparison of different authors, highlighting their main contributions, outcomes, and future potential. This table is a helpful resource for exploring studies related to traffic signal management.

Another innovative domain of traffic reduction is the utilization of adaptive traffic signal control (ATSC), a method proven to be highly effective in the field of traffic management. This methodology focuses on Transit Signal Priority (TSP), which involves modifying traffic signal cycles to reduce the time transit vehicles wait at red lights. Studies in this field primarily focus on implementing TSP-based ATSC to evaluate and reduce delays and queues in traffic flow. Table 2 provides a detailed analysis of how ATSC strategies are implemented for reducing traffic. This comparison highlights the varied contributions of several authors, demonstrating their unique techniques, significant discoveries, and influence on diminishing waits and queues. Highlighting Transit Signal Priority as a fundamental component emphasizes the importance of this flexible method in improving the effectiveness of traffic signal management systems.

The main obstacle is the ongoing incapacity to alleviate traffic congestion efficiently. Furthermore, the current methods have yet to be sufficient to produce the intended results. To tackle this complex problem, we have developed a new approach that uses RL-more precisely, the DQL algorithm. This method solves the long-standing problem and receives a great deal of historical attention, regularly piquing academics' curiosity in this emerging topic.

References	Key contribution	Result/findings	Future scope
Tan et al.[47]	Integrate a new reward function	Decrease queue length by up to 40% when compared to the baseline, which consists of deterministic and completely dynamic TSCs.	To expand the suggested schema for use in various intersections.
Li et al.[54]	Explored the application of the Single Architecture Ensemble (SAE) neural network architecture	Decreased average delay by up to 14% as compared to the initial measurement.	–
Wan et al.[48]	Explored the application of a dynamic reward factor	Decreased average delay by up to 20% compared to the initial measurement.	To expand the suggested schema for application in various intersections, alternative DRL approaches including actor-critic[55], deep deterministic policy gradient[56], and proximal policy optimization can be employed[57].
Chu et al.[58]	Explored the application of the LSTM neural network design.	Decreased average latency by 10% and queue length by 17% compared to the baseline.	To expand the suggested schema in order to enhance communication between multiple crossings.
Gender et al.[49]	Explored the utilization of extensive state space	Decreased average queue length by up to 30% compared to the initial measurement.	Extend the proposed schema to regulate red and yellow TSC.
Gong et al.[50]	Integrate MARL with the 3DQN architecture.	Decrease average queue length by up to 46% compared to the baseline.	To expand the suggested plan in order to guarantee equity among traffic streams.
Gao et al.[59]	Explored the utilization of an extensive state space	Decreased average delay by up to 29% in comparison to the initial measurement.	To expand the suggested schema for use in various intersections.
Wang et al.[60]	Examine the utilization of high-resolution event-based data	Decrease average latency by a maximum of 21% and queue length by up to 30% as compared to the initial measurement.	To expand the suggested schema to incorporate traffic disruptions such as detector noise, traffic accidents, and adverse weather conditions.
Mousavi et al.[51]	Contrast the value-based and PG-based approaches with a baseline model that utilizes a completely dynamic TSC based on a neural network with one hidden layer.	The PG-based solution decreases the average delay by a maximum of 43% and reduces the queue length by 40%.	To expand the suggested schema for use in various intersections.
Liang et al.[52]	implement a prioritized experience replay method	Decrease average waiting time by up to 20% compared to the initial measurement.	–
Van et al.[53]	Implement the max-plus coordination and transfer planning method.	Decrease average delay by up to 20% in comparison to the MARL.	To consider adopting various DRL methodologies due to the utilization of the conventional DQN methodology in the proposed method. Which has proven to be unstable
Wei et al.[61]	Studied the application of actual traffic datasets	Decreased average delay by a maximum of 19% and queue length by a maximum of 38% compared to the initial measurement.	To expand the suggested schema to regulate the yellow phase of TSC

Table 1. Comparison of conventional state of art methods for traffic system.

References	DRL method	Number of intersections	Compared against	Control strategy	Improvements
Lin et al.[62]	ResNetbasedA2C	9	ctuatedcontroller	Cyclic fixed TSP switch	16% lower Waiting time
Genders and Razavi[49]	DDQN + ER	1	STSCA	Acyclic TSPs with intermediate TSPs	82% lower Overall Delay
Chu et al.[58]	Stabilised IA2C	30	IA2C,IQL-LR,IQL-DNN	Acyclic TSPs with fixed duration	63.7% lower Average Delay
Li et al.[54]	DeepSAE + RL	1	Q-learning algorithm	Two TSPs with dynamic duration	14% lower Overall Delay
Yang et al.[63]	Regional A3C + PER 2	42	Rainbow DQL, Hierarchical MARL, Decentralized multi-agents	Acyclic TSP selection and TSP duration computation	8.78% lower Average Delay
Casas[64]	DDPG	43	Q-learning algorithm	Cyclic TSPs with computed duration	No data
Gao et al.[59]	DDQN + ER	1	LQF, fixed time controller	Cyclic TSPs with fixed duration	47% lower Overall Delay
Tan et al.[65]	Hierarchical regional A2C	24	Regional DRL	Acyclic TSPs with fixed duration	44.8% lower Waiting time
Wang et al.[60]	3DQN + PER	1	Actuated and fixed time controller	Acyclic TSPs with fixed duration	10.1% lower Average Delay

Table 2. Analyse and compare strategies used by ATSC to reduce traffic.

Research methodology

We have devised a method to address traffic congestion by employing DQN. In this section, we have provided the implementation of the DQL model for executing in traffic reduction.

Dataset analysis and discussion

Our approach used two XML datasets[66], both containing time series data. The first dataset contains environmental data, including vehicle ID, route, depart Lane, and depart Speed. This dataset is crucial for creating an intersecting environment consisting of 205 edges representing unique environment points used to train our agent. The second dataset focuses on route information and includes attributes such as edge ID, lane ID, index, length, shape, and function. These criteria are crucial for determining the paths within a junction. By effectively joining these two datasets, we could train our agent successfully. The model benefits from integrating environmental and route data, enhancing knowledge of system dynamics.

After the training, our model produced two extra datasets named "plot_queue_data" and "plot_reward_data". The datasets were crucial for the following testing step. We have assessed the wait length and rewards by analyzing the data produced by our trained agent during testing. The comprehensive testing procedure enabled us to evaluate the model's performance and effectiveness in real-world situations, offering vital insights about queue dynamics and rewards obtained by the agent.

We were evaluated 30 episodes during our testing phase using specific parameters. The maximum number of steps permitted was established at 240, while the quantity of cars produced for testing remained at 1000. The testing setup included a neural network structure of four layers with a learning rate 0.001. Our testing primarily focused on four specific actions: NSA (North-South Arms), NSLA (North-South Left Arms), EWA (East-West Arms), and EWLA (East-West Left Arms). The actions were intentionally selected to reduce queue length and maximize incentives in the system. This method enabled us to thoroughly assess the model's performance in many scenarios, giving us a detailed insight into it's skills in improving queue dynamics and reward results.

Reinforcement learning for traffic reduce

Implementing RL in traffic management can enhance signal timings and route allocations by adjusting to real-time traffic circumstances. This technique minimizes traffic and improves transportation efficiency by encouraging driving behavior and reducing congestion. The technology can adjust and improve traffic flow in urban areas by continuously learning from traffic patterns. Utilizing RL shows potential for developing intelligent traffic control systems that enhance sustainable and efficient urban mobility. Figure 1 displays the RL life cycle.

Here, S_t represents the state cycle of the environment, A_t is the action taken at time step t, and R_t is the reward associated with the possible action.

RL occurs in an environment where an agent learns through interaction and prediction of its actions. The favorable results of accurate predictions measure effectiveness, whereas less effective acts lead to increased penalty rates. This method encourages the agent to constantly improve decision-making by prioritizing activities that result in positive results and discouraging those with negative repercussions. RL adaptability enables the development of intelligent agents that can navigate their surroundings using more optimized tactics.

Agent training with environment

During the training of DQL agent, several crucial procedures are necessary for achieving success. We set up a replay memory to preserve events to facilitate effective learning from previous interactions. It is essential to provide a dynamic environment that reflects the difficulty of the actual world, offering the agent a variety of events to learn from. Establishing suitable parameters is crucial, as they determine the agent's behavior and learning speed. The initialization of the DQN is crucial for its learning process as it allows the network to improve its knowledge of optimal behaviors over time through experience. The agent uses a accurately designed loss function to enhance its decision-making skills. This function acts as a guiding principle for the agent to alter its weights and approximate optimal Q-values, thus improving its performance gradually. Figure 2 illustrates the Agent Training process using the DQL model.

Ahead of new knowledge, the agent uses the learning to negotiate the difficulty of the natural world, making informed and wise judgments that can result in the best possible outcomes. The DQL agent improves continually by learning and applying knowledge Repeated, becoming more skilled at solving real-world problems.

This study explores the complex interactions of transportation and urban traffic management to improve traffic flow, reduce congestion, and encourage sustainable mobility in metropolitan areas. The research analyzes the factors influencing traffic patterns and operational features on a complex road network with four principal directions. The study aims to get a thorough understanding of the behavior of the road network by conducting detailed analysis and observation, revealing its complexities and fundamental dynamics. The research tries to discover crucial jams, inefficiencies, and improvement opportunities inside the transportation system by comprehending its intricate relationships. The study intends to suggest new strategy and actions to improve traffic flow, reduce congestion, and enhance urban mobility solutions. The project aims to provide practical

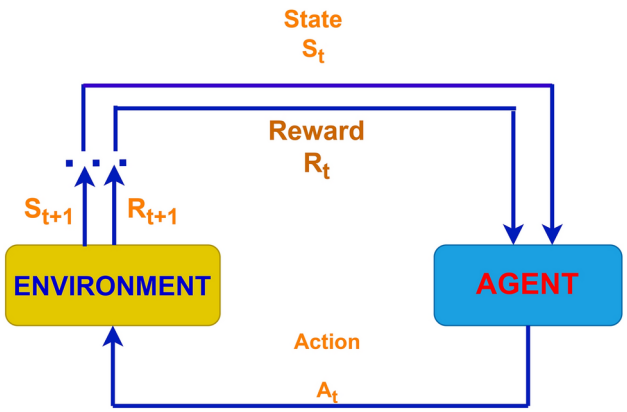

Fig. 1. The RL life cycle involves the agent learning from its environment, taking actions, and receiving rewards depending on its actions.

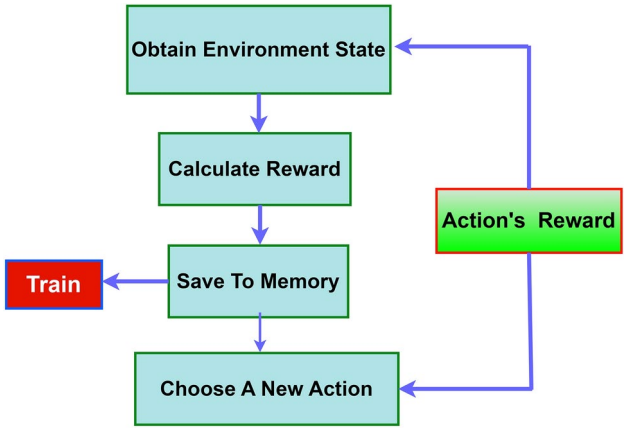

Fig. 2. RL involves defining execution criteria such as state, action, and rewards, as well as demonstrating the training process of the agent.

advice to policymakers and urban planners using a comprehensive strategy, including data analysis, modeling, and simulation. This will help create more dynamic, sustainable, and livable cities.

State of RL from intersection
A state depicts different node positions in the environment, where numerous nodes together indicate a state. Here, the state contains details regarding the positions of objects and node identifiers. The expression is:

$$S = (n_1, v_1, d_1, n_2, v_2, d_2, \cdots, n_i, v_i d_i) \tag{1}$$

Where n_i represents the number of vehicles, v_i denotes the velocity of the i_{th} vehicle, and d_i indicates the distance to the subsequent vehicle for each vehicle *i*.

Imagine lanes where the color green represents 1 and the color red represents 0.

$$\begin{bmatrix} G & G & G \\ R & G & R \\ R & R & R \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3}$$

Consider the 1st state:

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Since all nodes are set to 1, the active green signal allows cars to move in this state.

Furthermore, all nodes are red in the final state, rendering vehicle movement impossible.

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \tag{5}$$

Each state contains a unique value that determines the allowed motions of the vehicle according to the assigned value. Figure 3 effectively conveys the state through node and value representations. If the state node is present, it contains 1; if it is null, it contains 0.

Action of RL from intersection
In RL for traffic signal control, an "action" is a decision made by the RL agent on managing the traffic signals at a specific intersection. The RL agent learns to make decisions by observing the environment and aiming to optimize specific objectives, including lowering traffic congestion, minimizing delays, or enhancing traffic flow. Our system's potential actions include:

$$A = NSA, NSLA, EWA, EWLA$$

Every action the RL agent makes is associated with a particular arrangement of the traffic signal durations. The RL agent is given feedback in the form of rewards or penalties depending on the outcomes of its activities.

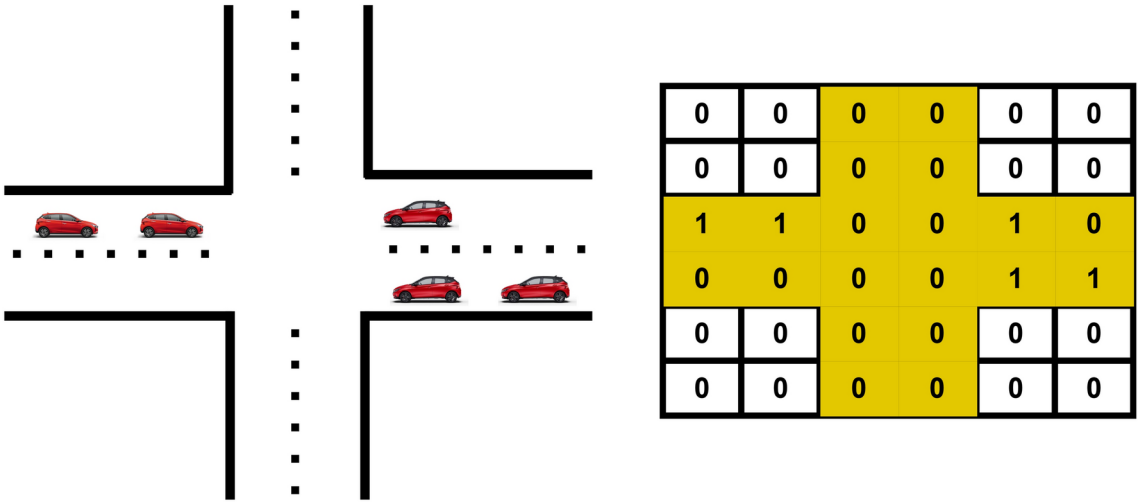

Fig. 3. A state representation where several edges indicate the presence or absence of a vehicle.

The RL agent aims to develop a policy that correlates observable traffic environment conditions with optimal behaviors to maximize cumulative rewards over time. We have worked at a junction with four lanes. Each lane consists of two blocks. Figure 4 illustrates the operation of our system, displaying various intersection directions.

Rewards of RL from intersection

The enhancement of overall traffic flow efficiency and safety characterizes the reward of a traffic reduction system at an intersection. The optimized coordination and smooth interaction of vehicles result in reduced congestion, minimized travel delays, and better vehicular throughput. The rewards system based on intersections creates a coordinated and efficiently managed traffic network that promotes a more sustainable and efficient urban environment for all individuals using the roads.

The function of rewards as expressed in the intersection is:

$$R(s, a, \acute{s}) \tag{6}$$

- s: This accurately represents the current environment. States provide the conditions or context in which the agent is working. The award may be specific to the current scenario.
- a: This signifies the actions the agent has completed up to this point. The agent's action determines the reward.
- s': This is the state resulting from the environmental transformation caused by an agent's action, denoted as a. The final state may also influence the rewards.Rewards are generally calculated based on the current environment s, the agent's action a, and the resulting environment s'. These three variables together express the reward during both training and testing. Rewards value may be negative. We have received the detrimental outcomes of our efforts. It articulates the consequences or sanctions employed to direct the learning process and influence the behavior of the RL agent. The agent learns to link some activities with negative consequences by introducing negative rewards, prompting it to modify its policy to reduce the incidence of undesirable behaviors.

Deep Q network implementation

During the Implementation of DQN, the agent discovers the most effective action through ongoing interaction with the environment, relying on a persistent trial-and-error approach. The ideal action is determined by considering both the immediate and potential rewards in the next index, up to n steps ahead. In the DQL algorithm, we were utilizef $V^{\pi}(S_i)$ to denote the total reward under a given policy π.

$$V^{\pi}(S_i) = r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + L \tag{7}$$

The DQL algorithm computes the Q value to assess a specific state and activity. This value is determined by considering both immediate and discounted rewards. The formulation of the Q value is as follows:

$$Q(s_i, a_i) \leftarrow r_i + \gamma V^{r}(S_{i+1}) \tag{8}$$

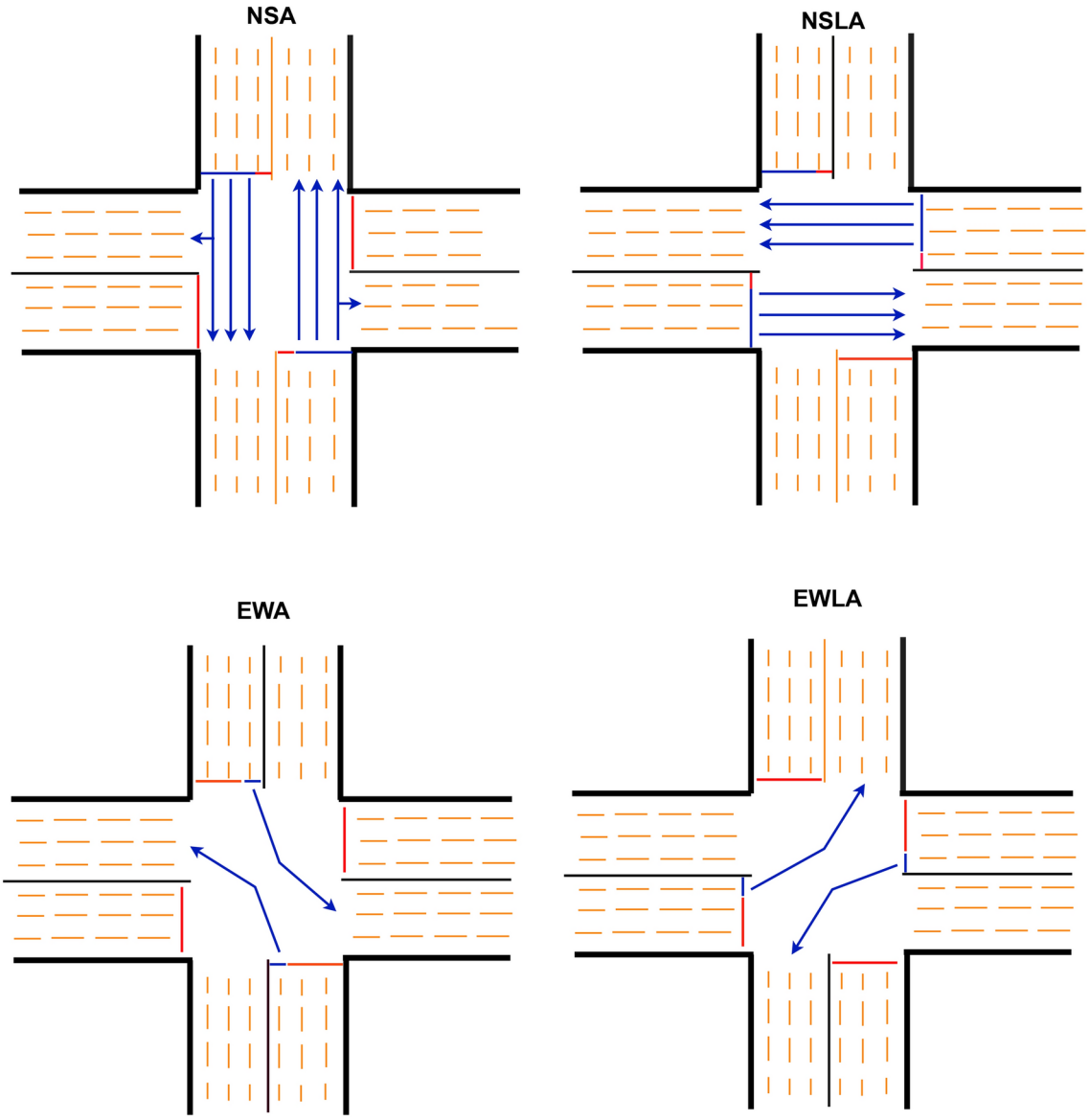

Fig. 4. Interaction at our intersection where many paths are depicted.

The discount coefficient, denoted by $\gamma\,(0 < \gamma < 1)$, represents the impact of future rewards on the current activity. The objective of Q learning is to optimize the overall utility. Therefore, in (8), we substitute the variables r_i and $V^{\pi}\,(S_{i+1})$ with the values O_i and max $Q_{a_{i+1},\in A}\,(S_{i+1}, a_{i+1})$, respectively.

$$Q\,(s_i, a_i) \leftarrow O_i + \gamma\, maxQ_{a_{i+1},\in A}\,(S_{i+1}, a_{i+1}) \tag{9}$$

Here, A represents the collection of action sets. The primary challenge in DQN during the learning phase is to effectively manage the trade-off between exploring new actions and exploiting the current knowledge of the action set. Specifically, in cases when the system is of significant size, the selection of the appropriate action will directly impact the convergence of the algorithm and the system's performance. Hence, we have incorporated an adjusted index value are same identify the optimal action to get the most effective action. The index value can accurately represent the changes in rewards and quickly modify the range of exploration to minimize the unnecessary selection cost.

$$a_i \leftarrow arg\max_{a}\,(Q\,(S_i, a)) + Index\,(s_i, a) \tag{10}$$

Q represents the assessment value assigned to the current state and activity. The introduction of $Index\,(s_i, a)$ is aimed at determining the optimal probable action based on the Q value. The expression evaluates to the value of (11). This formula provided is a type of index or score used in decision-making processes, often found in algorithms related to RL or optimization.

$$Index\left(S_i, a\right) = C_p\sqrt{2ln\, i \times min\left\{\frac{1}{4}V_a\left(i\right)\right\}/T_a\left(i\right)} \tag{11}$$

C_p is a positive constant, as shown by the notation 0[67]. T_a represents the frequency of selecting action after *i* frames. $V\left(i\right)_a$ represents the bias component, which incorporates the action's utility value variance $\sigma_a^2\left(i\right)$ to indicate its level of volatility. Below Table 3 showcase the index value calculation and model optimization criteria.

$$\sigma_a^2\left(i\right) = \sum_{k=1}^{T_a(i)} O^2\left(S_k, a\right)/T_a\left(i\right) - O^2\left(S_{T_a(i)}, a\right) \tag{12}$$

$$V_a\left(i\right) = \sigma_a^2\left(i\right) + \sqrt{2ln\, i/T_a\left(i\right)} \tag{13}$$

One approach to action selection is to use the action index, which considers the present action's system utility and gradually rank of actions with a more significant impact. This approach demonstrates the system's tendency to utilize its resources. Alternatively, in the continuous iterative process, if a particular action is not chosen or the chosen quantity is meager, there is a tendency to favor selecting that action in the subsequent iteration, demonstrating the explorative nature of the process.

Once the execution action a_i is determined, the relay carries out the operation by calculating the utility value *O* and updating the *Q* value using the following:

$$\begin{aligned} &Q_{i+1}\left(S_i, a_i\right) = \left(1-\alpha\right)Q_i\left(S_i, a_i\right) + \alpha\left(O_i + \gamma\, maxQ_i\left(S_{i+1}\right), a_{i+1}\right) \\ &\quad \text{if} \quad s = s_i\, and\, a = a_i \end{aligned} \tag{14}$$

The learning rate $\alpha\left(0 < \alpha \leq 1\right)$ of the state action is calculated as $\alpha = 1/\left(1 + T_a\left(i\right)\right)$.

Scenario	Parameters	Index value	Optimization implication
A	$C_p = 1, i = 5, V\left(i\right)_a = 2, T_a = 10$	0.40	Focus on exploitation due to frequent trials and moderate exploration.
B	$C_p = 2, i = 20, V\left(i\right)_a = 4, T_a = 4$	2.45	Higher emphasis on exploration due to lower action frequency and higher variance.
C	$C_p = 1.5, i = 50, V\left(i\right)_a = 1, T_a = 20$	0.67	Balanced but slightly more focused on exploitation due to high action frequency and lower variance.
D	$C_p = 3, i = 10, V\left(i\right)_a = 5, T_a = 2$	5.09	Strong focus on exploration due to high variance and low action frequency.

Table 3. Determine the index value calculation and optimization implication details.

Initialize replay memory $\mathscr{D}$ with capacity N
Initialize Q-network with weights θ
Initialize target Q-network with weights $\theta^- \leftarrow \theta$
Set exploration rate ε, discount factor γ, learning rate α, mini-batch size B, target network update frequency C
for episode = 1 to M **do**
  Initialize state s_1
  for t = 1 to T **do**
    With probability ε select a random action a_t
    Otherwise, select $a_t = \arg\max_a Q(s_t, a; \theta)$
    Execute action a_t, observe reward r_t and new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathscr{D}$
    Sample random mini-batch of transitions (s_i, a_i, r_i, s_{i+1}) from $\mathscr{D}$
    Compute target values: $y_i = \begin{cases} r_i & \text{for terminal } s_{i+1} \\ r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta^-) & \text{non-terminal } s_{i+1} \end{cases}$
    Update Q-network weights by minimizing the loss:

$$L(\theta) = \frac{1}{B} \sum_i (Q(s_i, a_i; \theta) - y_i)^2$$

    Every C steps, update target Q-network: $\theta^- \leftarrow \theta$
  end for
end for

Algorithm 1. Deep Q-Learning algorithm for Traffic Reduction

The Algorithm 1 express DQL sets up a replay memory $\mathcal{D}$ to save previous experiences, a Q-network with weights θ, and a target Q-network with weights θ^-. The goal of the Q-network is changed frequently to align with the present Q-network. several parameters are defined, such as the exploration rate (ϵ). discount factor γ, learning rate α, mini-batch size (B), and target network update frequency (C). These factors impact the learning process and should be adjusted according to the specific traits of the traffic environment. The algorithm progresses via episodes, each symbolizing a series of encounters with the traffic environment. The agent performs actions in the environment and then observes the states and rewards that follow inside each episode. At each time step, the agent chooses an action using an exploration strategy. An action is selected randomly with a probability of epsilon to explore the environment. The action with the highest Q-value for the current state is then chosen. The algorithm saves every transition (consisting of state, action, reward, and next state) in the replay memory. Random mini-batches of experiences are sampled from the replay memory during training. This aids in disrupting the temporal relationship between successive experiences and enhances the stability of the learning process. The Q-network is updated by minimizing the temporal difference error between the predicted Q-values and target Q-values. The goal values are calculated using the observed rewards and the highest Q-value for the subsequent state. The update procedure is determined by the loss function $L(\theta)$. The target Q-network is periodically adjusted to align with the present Q-network to improve the stability of the learning process. This gentle update aids in preventing oscillations and enhancing convergence.

In the DQN, the output replaces $maxQ_{a\acute{\in} A}(S_{t+1}, \acute{a})$ value. Experience replay (ER) is another crucial component of the general DQL process. The ER serves as the memory buffer for holding the experience tuples $\{s_t, a_t, r_t, s_{t+1}\}$ during the observation phase in the DQL learning process. The observation phase starts with the practical implementation of DQL and concludes when the ER reaches maximum capacity. Mini-batches can be sampled from the ER after the observation phase. Mini-batches are the input training sets for the Q-value Deep Neural Network model. The ER has a fixed size, so when a new experience tuple is added, the oldest stored tuple is discarded.

To calculate target Q-values, it is common practice to utilize a separate target DQL model. The target DQL model's weights are updated periodically by copying weights $\acute{\theta}$ from the Q-value DQL model, which changes weights during each learning iteration-ensuring a temporally static Q-value target in the target DNN model by maintaining the fixed weights for a set length of time. This avoids the issue of a shifting target and thereby stabilizes the DQL learning process. DQL efficiently reduces reward overestimation in noisy controlled contexts like traffic flows in complicated traffic networks, improving overall performance. Figure 5 displays the functional scheme of general DQL.

Hyperparameter tuning and novelty of DQL model

We have implemented a list of the hyperparameter tuning process for our DQL model. Hyperparameter tuning refers to selecting the optimal set of hyperparameters for a machine-learning model. Hyperparameters are configuration settings used to control the learning process and the structure of the model, which are set before

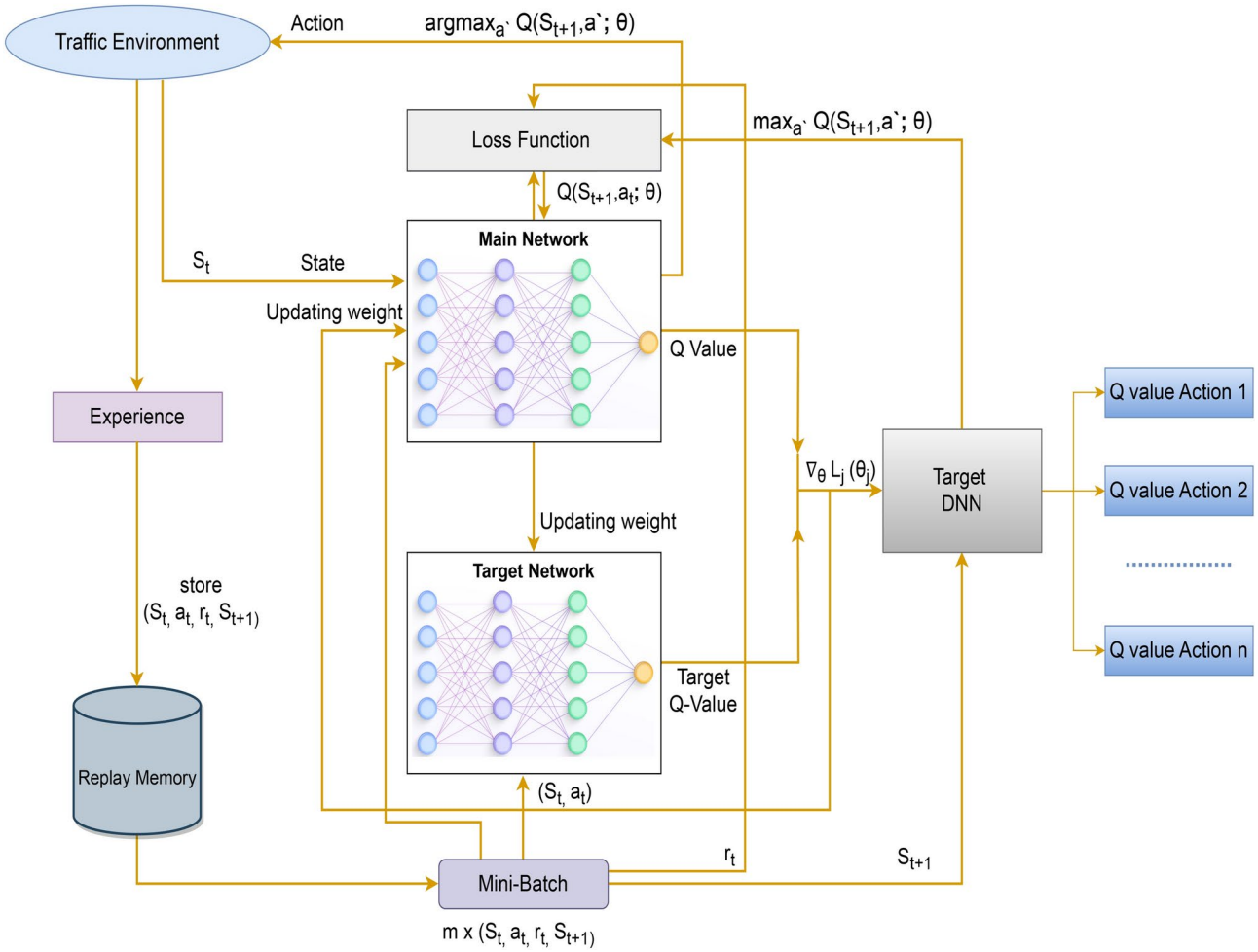

Fig. 5. The DQL Architecture proposes a model that expresses the working process and ultimately determines the appropriate action for traffic lanes.

the training process begins. Unlike the model parameters (e.g., weights in a neural network), they are not learned from the data but can significantly impact the model's performance. Hyperparameter tuning is needed because it helps optimize a machine learning model's performance by finding the best configuration of hyperparameters. Hyperparameters, set before the training process, significantly affect how well a model learns from data and performs on unseen data. Proper tuning can lead to better accuracy, faster convergence, and improved generalization, ensuring the model fits the training data well and performs effectively on new, unseen data. With tuning, models may avoid overfitting, where they learn the training data too well but fail to generalize, or underfitting, where they do not learn the data well enough. Thus, hyperparameter tuning is crucial for achieving the best possible model performance. We have analyzed many factors like learning rate, dropout, padding, optimizer, weight delay, batch size, epoch, activation, and kernel initializer. To perform hyperparameter tuning, start by defining the search space by selecting the hyperparameters and their possible values. Next, choose a strategy or algorithm. Split the dataset into training and validation sets, then train models using different combinations of hyperparameters. Evaluate each model's performance using a chosen metric, and finally, select the hyperparameters that yield the best validation performance. Here, we identify a specific range of parameter values, which constitutes the search space for the most optimal results in traffic reduction. Then, we select a value from this range to implement our system. Table 4 displays the outcomes of the Hyperparameter tuning process for our DQL model. By conducting methodical experiments, hyperparameter tweaking optimizes the model's performance, improving its ability to forecast outcomes and its resilience accurately.

DQL model has significant innovation in RL, particularly in the context of artificial intelligence and machine learning. This approach revolutionizes how agents learn to make decisions in complex environments by leveraging deep neural networks. The key innovation lies in its ability to combine Q-learning, a traditional RL technique, with deep neural networks to handle high-dimensional state spaces. DQL can effectively learn optimal strategies in environments with extensive and diverse state spaces by using neural networks to approximate the Q-function (which estimates the expected future rewards for taking a particular action in a given state). This innovation has successfully applied RL to a wide range of challenging tasks, autonomous driving, and robotic control. DQL represents a fundamental advancement in the field, opening doors to more sophisticated and capable AI

Parameter	Search space	Selected value
Total episodes	[15, 20, 30]	30
Max steps	[220, 230, 240]	240
n cars generated	[1000, 1150, 999]	1000
Width layers	[350, 400]	400
Num of layers	[4]	4
Batch size	[16, 32,8]	32
Learning rate	[0.001, 0.0001]	0.001
Training epochs	[800]	800
Num of states	[80]	80
Num of actions	[4]	4
Gamma	[0.80, 0.76, 0.75]	0.75

Table 4. Hyparameter tuning of proposed model.

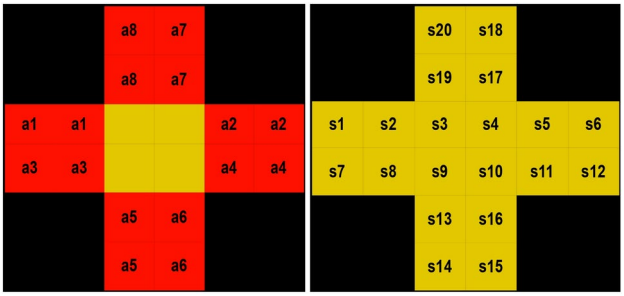

Fig. 6. Showcase the potential actions of a different lane and state representation at a junction.

systems that can learn directly from raw sensory inputs and navigate complex decision-making scenarios. A novel application of DQL in traffic management aims to significantly reduce congestion by optimizing traffic signal timings. This approach leverages the DQL algorithm's ability to learn and adapt to complex, dynamic environments. By integrating real-time traffic data, such as vehicle flow and occupancy sensors, the DQL model can predict and respond to traffic patterns with high precision. The model continuously updates its strategy based on traffic conditions, proactively adjusting signal timings to minimize jams and improve overall traffic flow. This innovative use of DQL enhances traffic efficiency, reduces travel times, and lowers emissions, addressing critical challenges in urban mobility.

Result analysis

Evaluation criteria and improvement

We carefully map the intersection's state, calculate the rewards, and then assign state names and actions to each lane sequentially. Figure 6 graphically depicts the various states of the intersection and the accompanying actions for distinct lanes. This model allows us to identify which action corresponds to each state accurately. Our approach focuses on recognizing the present state, predicting the next stage of the junction, and evaluating if the rewards are improving.

Our findings show that by identifying a vehicle's current condition, we may predict its likely future state and determine the most efficient subsequent state for the agent, thus improving overall performance. Table 5 lists essential states such as s1, s2, s7, s8, s6, s5, s12, s11, s14, s13, s16, s15, s20, s19, s18, and s17, showing them as current states and suggesting the best next states. This table displays the possible future states for vehicles based on their current status and the incentives linked to each state, highlighting the expected gain. This focus table is essential in our traffic reduction system, offering significant insights for optimizing the intersection's functionality.

The performance of our model showed great potential during both the training and testing stages. Within the training set, the queue duration experienced a decrease to 852, but the related incentives, negative at − 94492, indicate an enhanced comprehension and adjustment within the RL environment. The queue length decreased to 418 during the testing phase, while a positive testing reward of 8520 was obtained. These results jointly demonstrate a significant decrease in the length of the wait and an improvement in incentives compared to the initial state. This indicates the model's efficacy in acquiring knowledge and improving its performance, showcasing favorable advancements in training and testing situations.

$$queue_r = \frac{Testing\ queue\ length}{Training\ queue\ length} \times 100 \tag{15}$$

Current state	Next State	Action	Rewards
s1	(s2, s3, s4, s5, s6)	a2	++
s1	(s2, s3, s10, s16, s15)	a6	++
s1	(s2, s3, s19, s20)	a8	++
s2	(s3, s4, s5, s6)	a2	++
s2	(s3, s10, s16, s15)	a6	++
s2	(s3, s19, s20)	a8	++
s7	(s8, s9, s10, s11, s12)	a4	++
s7	(s8, s9, s4, s17, s18)	a7	++
s7	(s8, s9, s13, s14)	a5	++
s8	(s9, s10, s11, s12)	a4	++
s8	(s9, s4, s17, s18)	a7	++
s8	(s9, s13, s14)	a5	++
s6	(s5, s4, s3, s2, s1)	a1	++
s6	(s5, s4, s9, s13, s14)	a5	++
s6	(s5, s4, s17, s18)	a7	++
s5	(s4, s3, s2, s1)	a1	++
s5	(s4, s9, s13, s14)	a5	++
s5	(s4, s17, s18)	a7	++
s12	(s11, s10, s9, s8, s7)	a3	++
s12	(s11, s10, s16, s15)	a6	++
s12,	(s11, s10, s3, s19, s20)	a8	++
s11	(s10, s9, s8, s7)	a3	++
s11	(s10, s16, s15)	a6	++
s11	(s10, s3, s19, s20)	a8	++
s14	(s13, s9, s8, s7)	a3	++
s14	(s13, s9, s3, s19, s20)	a8	++
s14	(s13, s9, s4, s5, s6)	a2	++
s13	(s9, s8, s7)	a2	++
s13	(s9, s3, s19, s20)	a8	++
s13	(s9, s4, s5, s6)	a2	++
s15	(s16, s10, s11, s12)	a4	++
s15,	(s16, s10, s4, s17, s18)	a7	++
s15	(s16, s10, s3, s2, s1)	a1	++
s16	(s10, s11, s12)	a4	++
s16	(s10, s4, s17, s18)	a7	++
s16	(s10, s3, s2, s1)	a1	++
s20	(s19, s3, s2, s1)	a1	++
s20	(s19, s3, s9, s13, s14)	a5	++
s20	(s19, s3, s10, s11, s12)	a4	++
s19	(s3, s2, s1)	a1	++
s19	(s3, s9, s13, s14)	a5	++
s19	(s3, s10, s11, s12)	a4	++
s18	(s17, s4, s5, s6)	a2	++
s18	(s17, s4, s10, s16, s15)	a6	++
s18	(s17, s4, s9, s8, s7)	a3	++
s17	(s4, s5, s6)	a2	++
s17	(s4, s10, s16, s15)	a6	++
s17	(s4, s9, s8, s7)	a3	++

Table 5. Determine the current state and identify the next ideal state with increased rewards.

$$rewards_m = \frac{Testing\,reward}{Training\,rewards} \times 100 \tag{16}$$

In this context, $queue_r$ represents the decrease in queue length, while $rewards_m$ represents the maximization of outcomes.

Testing label	Findings (%)	Status of label
Queue length	49	Reduced
Rewards	9	Increased

Table 6. Enhancement of our system, which demonstrates a shorter wait time and more rewards.

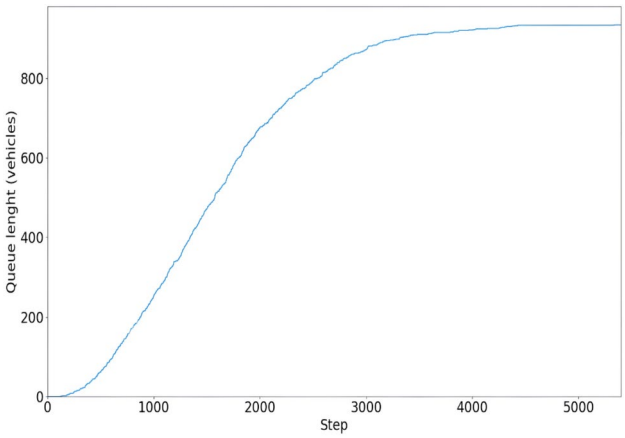

Fig. 7. The agent generates the queue length during training; this graph shows the queue length across steps.

We have achieved a notable reduction of 49% in queue length and a 9% increase in rewards. The results highlight the model's effectiveness in reducing the queue length by almost 50%, which is a significant improvement. This result is consistent with the formula for reducing traffic congestion, indicating its successful implementation. The significant reduction in line duration and simultaneous increase in incentives confirm the model's success in alleviating traffic congestion, demonstrating our strategy's effectiveness. In Table 6, we depict the percentage of progress about queue duration and rewards.

Training phase results in RL environment

We have successfully executed 5400 steps throughout our RL agent's training iteration. Significantly, the queue length reached a final index of 852 at the end of this training period. It is crucial to emphasize that the training data included information from the environment, edge, and vehicle sources. The 5400 steps provided extensive information about the edge information of the environment. Figure 7 visually depicts the training progression by demonstrating the changes in the queue length throughout the steps.

During the training phase, our model allocated rewards according to the queue length performance, which served as a metric to measure the success of the training process. The reward value received, − 944992, indicates a negative reward, suggesting a punitive factor in the setting of RL. Negative cultivate rewards are a type of feedback mechanism used to discourage undesirable behaviors by penalizing them. Unlike traditional negative rewards that reduce the agent's overall score, negative cultivate rewards focus on promoting better behavior by progressively penalizing actions that lead to less optimal outcomes. This approach can optimize traffic flow in traffic management by penalizing actions or behaviors that cause congestion or delays. Our traffic control system uses negative rewards to disincentivize actions like abrupt lane changes or excessive speeding, thereby encouraging smoother and more efficient traffic patterns. By applying these penalties, the system can gradually refine its strategies to minimize traffic jams and improve overall flow. Figure 8 illustrates the training rewards of our DQL model, showing the link between rewards and epochs. It provides insights into how the effectiveness of the training process changes over time.

Testing phase results in RL environment

After completing 30 epochs of testing on our DQL model, we have achieved a testing queue length of 418. This result indicates a successful reduction in the queue length compared to the queue during training. The lowered queue length indicates the reduction of traffic jams, demonstrating the efficacy of our methodology. The length of the queue during the testing phase is 418 (last index). Figure 9 graphically illustrates the length of the testing queue throughout several epochs, offering a distinct portrayal of the enhancements made in queue management. Our system significantly reduced queue length by almost 50% from training to testing. During the training phase, the last index queue value was 852; in the testing phase, it dropped to 418. This substantial reduction in queue length during testing indicates our system's effectiveness in alleviating traffic congestion. Thus, the testing results demonstrate the system's ability to reduce queue lengths and mitigate traffic congestion.

During the rewards calculation phase of the testing, we have considered 535 steps. This procedure measures the length of the queue, which indicates the amount of reward that the training receives. In this framework,

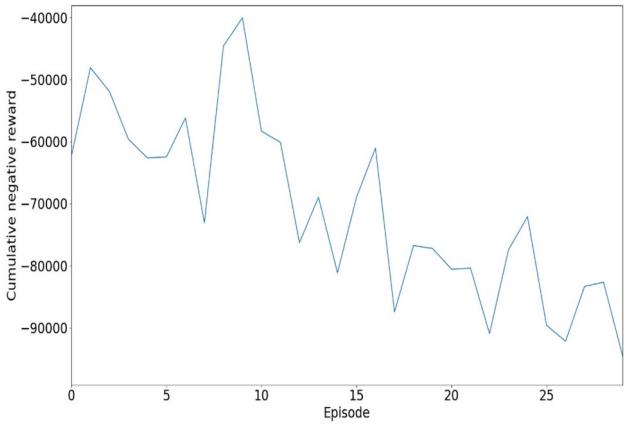

Fig. 8. The agent generates the rewards during training; this graph shows the rewards across steps.

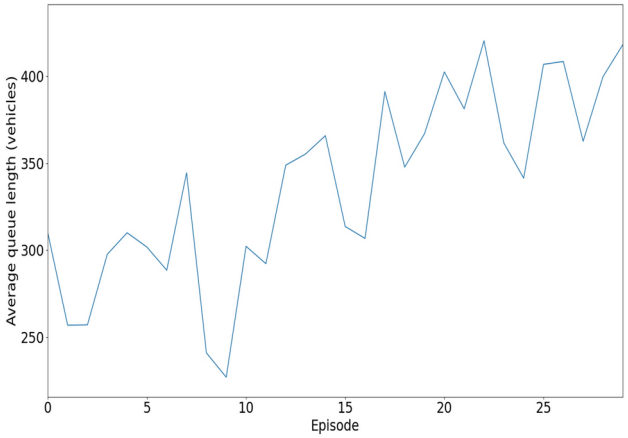

Fig. 9. Evaluating the testing queue length representation throughout episodes using the suggested model.

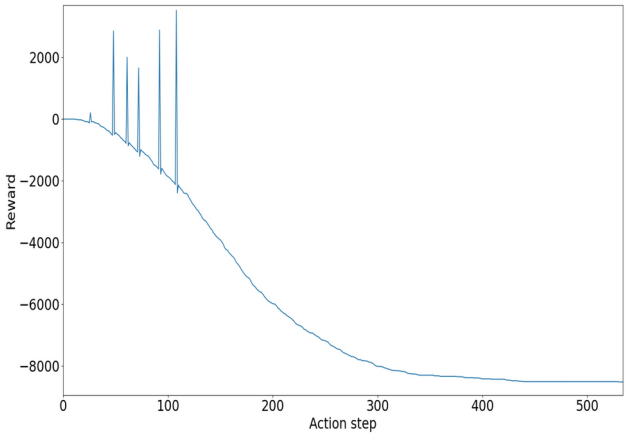

Fig. 10. Assessing the testing rewards based on the action step from the model.

negative rewards indicate a substantial penalty for the training, with smaller negative values indicating an enhancement in rewards. The incentives amount to − 8520.0 and are contained in the last action steps. Figure 10 illustrates the correlation between the rewards of the testing data and the action steps taken, presented in a visual format. During training, the final reward index was − 944992, whereas in testing, it improved significantly to − 8520. This substantial gap highlights a significant enhancement in performance. The rewards are maximized

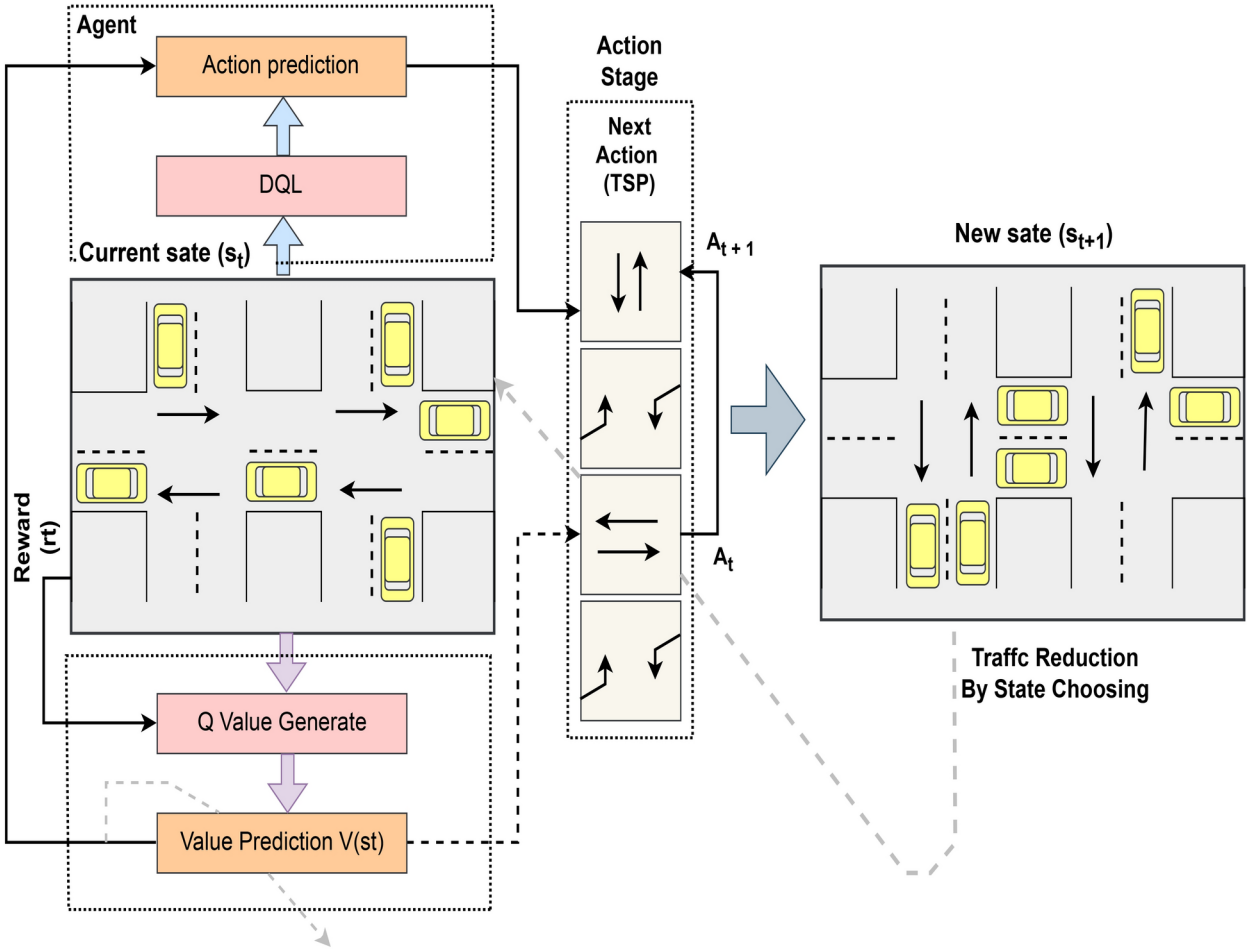

Fig. 11. The traffic reduction system involves finding the next ideal condition based on specific criteria.

during the testing phase, suggesting that the agent can make more effective decisions based on the testing rewards. This indicates that the agent's decision-making strategy has improved significantly when evaluated in the testing environment.

Effectiveness and findings of this research

This study presents an innovative traffic reduction system tailored for urban areas. Leveraging RL techniques, notably an advanced DQL algorithm, we have developed a system that operates on a reward-based mechanism. The premise is simple: when the queue length decreases, the agent gives a reward accordingly, incentivizing traffic optimization. Our results expresses the volumes about the efficacy of this approach. We have achieved a remarkable 49% reduction in queue length, translating to nearly a 50% decrease in waiting times-an impressive feat in traffic management. This substantial reduction in the queue length during the testing phase indicates a notable enhancement in system efficiency. By decreasing the queue length, our system demonstrated its capability to handle tasks more effectively, directly correlating to reducing traffic congestion. The implications of this improvement are significant. A shorter queue length suggests that our system can manage and process tasks faster and more efficiently. This improvement is crucial in real-world applications where reducing traffic congestion can enhance productivity, lower operational costs, and improve user satisfaction. This outcome underscores the effectiveness of our approach and validates our system's ability to alleviate traffic congestion. We can expect even greater efficiencies and performance improvements as we continue to refine and optimize our system. Moreover, our system demonstrates a 9% improvement in reward at every state, indicating consistent progress and refinement. The goal is to maximize cumulative rewards, which can be effectively applied to optimize traffic management and decision-making. To achieve this, an RL agent is trained to interact with a traffic simulation environment where it can take actions such as adjusting traffic signals, controlling traffic flow, or managing road usage. The agent receives feedback in the form of rewards based on the effectiveness of its actions in reducing congestion, minimizing delays, and improving overall traffic efficiency. Through exploration and exploitation, the agent learns to make decisions that balance immediate rewards with long-term benefits, ultimately optimizing traffic flow. By continually refining its strategy through trial and error, the RL agent can identify patterns and implement strategies that reduce traffic congestion and smooth traffic flow. This approach helps manage traffic in real-time and adapts to changing conditions and varying traffic patterns, ensuring that

decisions remain optimal under different scenarios. Table 5 encapsulates the key components of our approach, outlining the state-action relationships that drive our system's behavior. We have showcased our training data in Figures 7 and 8 to provide a comprehensive view of our methodology. These figures illustrate the evolution of queue length and associated rewards over the training period, showcasing the iterative refinement of our model. During testing, our system continues to demonstrate its prowess. Figures 9 and 10 present the outcomes of our test runs, highlighting the tangible reductions in queue length and the corresponding increase in rewards, further validating the effectiveness of our approach. Through rigorous analysis, we have observed a consistent reduction in queue length and a proportional increase in rewards across various states. This experiential evidence underscores the viability and impact of our traffic reduction system within urban environments.

Queue length reduction strategies can significantly impact traffic reduction by improving overall flow and decreasing congestion. Implementing measures such as optimizing traffic signal timings, enhancing lane management, and deploying ITS can reduce queue lengths. These strategies enhance efficiency and reduce commuter travel times by minimizing delays at intersections and jam areas. The effectiveness of such measures is notable in urban areas, leading to smoother traffic flow, decreased emissions from idling vehicles, and improved overall road safety. However, the success of these strategies often depends on comprehensive planning, coordination with local authorities, and continual monitoring and adjustments based on traffic patterns.

Discussion

We have created a novel system designed to decrease significantly traffic congestion at intersections by utilizing a complex method based on Deep DQL in an RL. Implementing DQN in a traffic reduction system offers the unique advantage of continuously improving traffic flow efficiency through real-time adaptive learning. Unlike traditional static algorithms, DQL can dynamically adjust traffic signals based on current conditions, learning from past traffic patterns and behaviors to optimize future decisions. This leads to reduced congestion, shorter travel times, and lower emissions as the system becomes increasingly adept at managing varying traffic volumes and unexpected disruptions, ultimately enhancing urban mobility and environmental sustainability. The technology begins by training an agent in a accurately designed environment replicating real-world intersection situations. The agent in this setting learns to evaluate the current condition of the intersection by analyzing whether vehicles are present or not in particular lanes. The agent analyses the states to find the best vehicle route, aiming to reduce queue duration and improve traffic flow. Our solution is centered upon a neural schema consisting of many nodes, each indicating the occupancy status of a lane (0 for empty, 1 for occupied). The agent uses the DQL model to assess the values linked to the nodes, producing Q-values that direct decision-making toward optimal traffic management solutions. Upon obtaining the current state (st), the agent strategically plans the vehicle trajectory by determining the next state (St+1) that maximizes traffic flow and minimizes congestion. This technique entails carefully calculating node values to evaluate possible actions and identify the most effective action. Additionally, our system views queue durations as a vital parameter for reducing traffic. For training our model, we focus on several specific parameters: total_episodes, max_steps, num_layers, width_layers, batch_size, learning_rate, training_epochs, num_states, num_actions, and gamma. We set each parameter to its optimal value to ensure the most effective training of our agent. To test our model, we focus on several parameters: max_steps, episode_seed, num_states, and num_actions. We set each parameter's optimal value to create the most effective traffic reduction system. These testing parameters are executed to compute the queue length and rewards. Based on the testing results, the agent can make decisions regarding the environment. The parameters chosen for this model reflect several advancements over previous works. Increasing the num_layers and width_layers enhances the network's depth and capacity, allowing it to better capture and represent complex patterns compared to shallower or narrower architectures used in earlier studies. Adjustments in batch_size and learning_rate improve the stability and efficiency of training; a larger batch size and a more adaptive learning rate schedule can lead to smoother convergence and better generalization, addressing the training issue found in prior methods. Additionally, extending total_episodes and training_epochs offers the model more exposure to diverse scenarios, mitigating underfitting problems encountered before. Changes in 'gamma' refine how future rewards are valued, potentially improving the model's alignment with the problem's dynamics and enhancing decision-making over the long term. Overall, these configurations aim to build on past limitations by offering a more robust and efficient learning process, leading to improved model performance and reliability in both training and testing phases. When a route has a low queue length, showing smoother traffic flow, the agent gives it priority over congested alternatives. Updating Q-values requires forecasting the future condition of cars, allowing the agent to enhance its comprehension of the best traffic control tactics. The system evolves continuously through an iterative process, dynamically adjusting to changing traffic conditions to maximize efficiency at intersections. During the last stage of our system, the agent distributes rewards according to its selected actions, a crucial step in encouraging efficient traffic control. The agent receives the most rewards by choosing the most efficient lane and demonstrating successful intersection navigation. If the selected lane is not as successful, the benefits decrease over time, prompting the agent to seek out better ways. Our solution continuously demonstrated superior queue lengths and rewards through thorough testing, highlighting its effectiveness in reducing traffic congestion. Figure 11 displays the comprehensive traffic reduction system, with each state readily visible graphically. The model was validated after an extensive investigation. An Intel(R) Core(TM) i7 CPU, 16GB RAM, and 12 GB GPU was used for the entire training procedure on a Windows 10 computer. TensorFlow 2.2.1 and Python 3.12.3 implemented all offensive automatic traffic reduction models. Python libraries such as TensorFlow, frequently used to create image classification models, may be managed more easily with the help of Spider. The findings confirm the system's effectiveness and create a significant historical record of traffic patterns, providing essential insights for improving urban mobility. Our technology

shows the potential to revolutionize urban traffic management by effectively combining reward mechanisms with intelligent decision-making. Our technology can adjust to changing traffic conditions and choose the best routes, leading to a more efficient and sustainable urban environment.

Conclusion and future work

RL gives substantial benefits in the application of transportation systems, where real-time adaptive control is critical to increasing efficacy and efficiency. Traffic Control approaches that rely on prespecified models of these processes are perceived to have a substantial disadvantage compared to the ability to learn through dynamic interaction with the environment. This paper introduces an innovative RL technique utilizing the DQL algorithm to minimize traffic congestion effectively. The system is structured based on an intersection-centered traffic model, emphasizing its ability to optimize waiting times and improve reward systems. This study's findings represent a significant advancement in traffic management, creating an effective method for decreasing traffic congestion. Our current method effectively manages road intersections and makes optimal decisions to reduce traffic congestion. This advancement shows significant potential and is a crucial addition to traffic control.

In the future, our model will be enhanced to work with real-time traffic data and optimization. This development will enable our system to connect to the internet, allowing the model to receive real-time data. In this capability, the agent can make informed decisions and adopt the optimal lane for vehicle movement. To address the high computational complexity in DQL for large-scale traffic networks, first, feature extraction, and dimensionality reduction techniques will reduce the state and action space. Secondly, more efficient neural network architectures will be used to improve processing efficiency. Additionally, techniques such as experience replay and target networks will stabilize learning and reduce redundant computations. Parallel computing and distributed learning will also be utilized to manage large-scale data by distributing the computational load across multiple processors, thereby cutting computational costs.

Data availibility

Data is available in a publicly accessible link: https://github.com/masfiq100/A-Reinforcement-Learning-Approach-for-Reducing-Traffic-Congestion-using-Deep-Q-Learning/tree/main/traffic/TLCS/intersection.

Received: 13 May 2024; Accepted: 7 October 2024
Published online: 12 December 2024

References

1. Qadri, S. S. S. M., Gökçe, M. A. & Öner, E. State-of-art review of traffic signal control methods: Challenges and opportunities. *Eur. Transp. Res. Rev.* **12**, 1–23 (2020).
2. Swapno, S. M. R. *et al.* An adaptive traffic signal management system incorporating reinforcement learning. In *2023 Annual international conference on emerging research areas: International conference on intelligent systems (AICERA/ICIS)*, 1–6 (IEEE, 2023).
3. Eom, M. & Kim, B.-I. The traffic signal control problem for intersections: A review. *Eur. Transp. Res. Rev.* **12**, 1–20 (2020).
4. Lee, W.-H. & Chiu, C.-Y. Design and implementation of a smart traffic signal control system for smart city applications. *Sensors* **20**, 508 (2020).
5. Swapno, S. M. R. *et al.* Traffic light control using reinforcement learning. In *2024 international conference on integrated circuits and communication systems (ICICACS)*, 1–7 (IEEE, 2024).
6. Wang, T., Zhu, Z., Zhang, J., Tian, J. & Zhang, W. A large-scale traffic signal control algorithm based on multi-layer graph deep reinforcement learning. *Transp. Res. Part C: Emerg. Technol.* **162**, 104582 (2024).
7. Ahmed, T., Liu, H. & Gayah, V. V. Identification of optimal locations of adaptive traffic signal control using heuristic methods. *Int. J. Transp. Sci. Technol.* **13**, 122–136 (2024).
8. Zhang, Z., Guo, M., Fu, D., Mo, L. & Zhang, S. Traffic signal optimization for partially observable traffic system and low penetration rate of connected vehicles. *Comput.-Aided Civil Infrastruct. Eng.* **37**, 2070–2092 (2022).
9. Wang, K., Shen, Z., Lei, Z. & Zhang, T. Towards multi-agent reinforcement learning based traffic signal control through spatio-temporal hypergraphs. arXiv preprint arXiv:2404.11014 (2024).
10. Al-Otaibi, S., Rehman, A., Mujahid, M., Alotaibi, S. & Saba, T. Efficient-gastro: Optimized efficientnet model for the detection of gastrointestinal disorders using transfer learning and wireless capsule endoscopy images. *PeerJ Comput. Sci.* **10**, e1902 (2024).
11. Mehmood, A. *et al.* Prosperous human gait recognition: An end-to-end system based on pre-trained CNN features selection. *Multimedia Tools Appl.* 1–21 (2020).
12. Hussain, N. *et al.* A deep neural network and classical features based scheme for objects recognition: An application for machine inspection. *Multimedia Tools Appl.* 1–23 (2020).
13. Hameed, U. et al. A deep learning approach for liver cancer detection in CT scans. *Comput. Methods Biomech. Biomed. Eng.: Imaging Visualiz.* **11**, 2280558 (2024).
14. Atteia, G. et al. Adaptive dynamic dipper throated optimization for feature selection in medical data. *Comput. Mater. Continua* **75**, 1883–1900 (2023).
15. Emary, E., Zawbaa, H. M., Hassanien, A. E., Schaefer, G. & Azar, A. T. Retinal blood vessel segmentation using bee colony optimisation and pattern search. In *2014 international joint conference on neural networks (IJCNN)*, 1001–1006, https://doi.org/10.1109/IJCNN.2014.6889856 (2014).
16. Ajeil, F. H., Ibraheem, I. K., Azar, A. T. & Humaidi, A. J. Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment. *Int. J. Adv. Rob. Syst.* **17**, 1729881420929498 (2020).
17. Kumar, V. *et al.* Recognition of Parkinson's disease using different machine learning models. In *2023 international conference on new frontiers in communication, automation, management and security (ICCAMS)*, vol. 1, 1–6 (IEEE, 2023).
18. Najm, A. A., Ibraheem, I. K., Azar, A. T. & Humaidi, A. J. Genetic optimization-based consensus control of multi-agent 6-DoF UAV system. *Sensors* **20**, 3576 (2020).
19. Koubâa, A., Ammar, A., Alahdab, M., Kanhouch, A. & Azar, A. T. Deepbrain: Experimental evaluation of cloud-based computation offloading and edge computing in the internet-of-drones for deep learning applications. *Sensors* **20**, 5240 (2020).
20. Aurangzeb, K. et al. Human behavior analysis based on multi-types features fusion and Von Nauman entropy based features reduction. *J. Med. Imaging Health Inf.* **9**, 662–669 (2019).
21. Rehman, A., Raza, A., Alamri, F. S., Alghofaily, B. & Saba, T. Transfer learning-based smart features engineering for osteoarthritis diagnosis from knee x-ray images. *IEEE Access* **11**, 71326–71338 (2023).

22. Swapno, S. M. R. *et al.* A novel machine learning approach for fast and efficient detection of mango leaf diseases. In *2024 IEEE 3rd international conference on computing and machine intelligence (ICMI)*, 1–7 (IEEE, 2024).
23. Rehman, A. et al. RDET stacking classifier: A novel machine learning based approach for stroke prediction using imbalance data. *PeerJ Comput. Sci.* **9**, e1684 (2023).
24. Ul Islam Khan, M. et al. Securing electric vehicle performance: Machine learning-driven fault detection and classification. *IEEE Access* **12**, 71566–71584 (2024).
25. Perveen, S. et al. Handling irregularly sampled longitudinal data and prognostic modeling of diabetes using machine learning technique. *IEEE Access* **8**, 21875–21885 (2020).
26. Mahmood, T., Mehmood, Z., Shah, M. & Saba, T. A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform. *J. Vis. Commun. Image Represent.* **53**, 202–214 (2018).
27. Nobel, S. N., Swapno, S. M. R., Islam, M. B., Meena, V. & Benedetto, F. Performance improvements of machine learning-based crime prediction, a case study in bangladesh. In *2024 IEEE 3rd international conference on computing and machine intelligence (ICMI)*, 1–7 (IEEE, 2024).
28. Yang, Z.-s., Chen, X., Tang, Y.-s. & Sun, J.-p. Intelligent cooperation control of urban traffic networks. In *2005 international conference on machine learning and cybernetics*, vol. 3, 1482–1486 (IEEE, 2005).
29. Srinivasan, D., Choy, M. C. & Cheu, R. L. Neural networks for real-time traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **7**, 261–272 (2006).
30. Wiering, M. A. *et al.* Multi-agent reinforcement learning for traffic light control. In *Machine learning: Proceedings of the seventeenth international conference (ICML'2000)*, 1151–1158 (2000).
31. Steingrover, M. *et al.* Reinforcement learning of traffic light controllers adapting to traffic congestion. In *BNAIC*, 216–223 (2005).
32. Steingrover, M. *et al.* Reinforcement learning of traffic light controllers adapting to traffic congestion. In *BNAIC*, 216–223 (2005).
33. Isa, J., Kooij, J., Koppejan, R. & Kuijer, L. Reinforcement learning of traffic light controllers adapting to accidents. *Design and Organisation of Autonomous Systems* (2006).
34. Abdoos, M., Mozayani, N. & Bazzan, A. L. Traffic light control in non-stationary environments based on multi agent Q-learning. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*, 1580–1585 (IEEE, 2011).
35. Bellavista, P., Caselli, F. & Foschini, L. Implementing and evaluating v2x protocols over itetris: traffic estimation in the colombo project. In *Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications*, 25–32 (2014).
36. Nobel, S. N. et al. A machine learning approach for vocal fold segmentation and disorder classification based on ensemble method. *Sci. Rep.* **14**, 14435 (2024).
37. Zhang, R. *et al.* Increasing traffic flows with dsrc technology: Field trials and performance evaluation. In *IECON 2018-44th annual conference of the IEEE industrial electronics society*, 6191–6196 (IEEE, 2018).
38. Nobel, S. N. et al. SegX-Net: A novel image segmentation approach for contrail detection using deep learning. *PLoS One* **19**, e0298160 (2024).
39. Tonguz, O. K. & Zhang, R. Harnessing vehicular broadcast communications: DSRC-actuated traffic control. *IEEE Trans. Intell. Transp. Syst.* **21**, 509–520 (2019).
40. Wang, F. et al. A group-based signal timing optimization model considering safety for signalized intersections with mixed traffic flows. *J. Adv. Transp.* **2019**, 2747569 (2019).
41. Stevanovic, J., Stevanovic, A., Martin, P. T. & Bauer, T. Stochastic optimization of traffic control and transit priority settings in VISSIM. *Transp. Res. Part C: Emerg. Technol.* **16**, 332–349 (2008).
42. Zhang, L., Yin, Y. & Chen, S. Robust signal timing optimization with environmental concerns. *Transp. Res. Part C: Emerg. Technol.* **29**, 55–71 (2013).
43. McKenney, D. & White, T. Distributed and adaptive traffic signal control within a realistic traffic simulation. *Eng. Appl. Artif. Intell.* **26**, 574–583 (2013).
44. Boillot, F., Midenet, S. & Pierrelee, J.-C. The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transp. Res. Part C: Emerg. Technol.* **14**, 18–38 (2006).
45. Cai, C., Wong, C. K. & Heydecker, B. G. Adaptive traffic signal control using approximate dynamic programming. *Transp. Res. Part C: Emerg. Technol.* **17**, 456–474 (2009).
46. Mirchandani, P. & Head, L. A real-time traffic signal control system: architecture, algorithms, and analysis. *Transp. Res. Part C: Emerg. Technol.* **9**, 415–432 (2001).
47. Tan, K. L., Poddar, S., Sarkar, S. & Sharma, A. Deep reinforcement learning for adaptive traffic signal control. In *Dynamic systems and control conference*, vol. 59162, V003T18A006 (American Society of Mechanical Engineers, 2019).
48. Wan, C.-H. & Hwang, M.-C. Value-based deep reinforcement learning for adaptive isolated intersection signal control. *IET Intel. Transp. Syst.* **12**, 1005–1010 (2018).
49. Genders, W. & Razavi, S. Using a deep reinforcement learning agent for traffic signal control. arXiv preprint arXiv:1611.01142 (2016).
50. Gong, Y., Abdel-Aty, M., Cai, Q. & Rahman, M. S. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning. *Transp. Res. Interdiscip. Perspect.* **1**, 100020 (2019).
51. Mousavi, S. S., Schukat, M. & Howley, E. Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intel. Transp. Syst.* **11**, 417–423 (2017).
52. Liang, X., Du, X., Wang, G. & Han, Z. Deep reinforcement learning for traffic light control in vehicular networks. arXiv preprint arXiv:1803.11115 (2018).
53. Van der Pol, E. & Oliehoek, F. A. Coordinated deep reinforcement learners for traffic light control. *Proceedings of learning, inference and control of multi-agent systems (at NIPS 2016)*, Vol. 8, 21–38 (2016).
54. Li, L., Lv, Y. & Wang, F.-Y. Traffic signal timing via deep reinforcement learning. *IEEE/CAA J. Automatica Sinica* **3**, 247–254 (2016).
55. Konda, V. & Tsitsiklis, J. Actor-critic algorithms. *Adv. Neural Inf. Process. Syst.*, **12** (1999).
56. Chu, X. & Ye, H. Parameter sharing deep deterministic policy gradient for cooperative multi-agent reinforcement learning. arXiv preprint arXiv:1710.00336 (2017).
57. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).
58. Chu, T., Wang, J., Codecá, L. & Li, Z. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Trans. Intell. Transp. Syst.* **21**, 1086–1095 (2019).
59. Gao, J., Shen, Y., Liu, J., Ito, M. & Shiratori, N. Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network. arXiv preprint arXiv:1705.02755 (2017).
60. Wang, S., Xie, X., Huang, K., Zeng, J. & Cai, Z. Deep reinforcement learning-based traffic signal control using high-resolution event-based data. *Entropy* **21**, 744 (2019).
61. Wei, H., Zheng, G., Yao, H. & Li, Z. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2496–2505 (2018).
62. Lin, Y., Dai, X., Li, L. & Wang, F.-Y. An efficient deep reinforcement learning model for urban traffic control. arXiv preprint arXiv:1808.01876 (2018).
63. Yang, S., Yang, B., Wong, H.-S. & Kang, Z. Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm. *Knowl.-Based Syst.* **183**, 104855 (2019).

64. Casas, N. Deep deterministic policy gradient for urban traffic light control. arXiv preprint arXiv:1703.09035 (2017).
65. Tan, T. et al. Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE Trans. Cybern.* **50**, 2687–2700 (2019).
66. A Reinforcement Learning Approach for Reducing Traffic Congestion using Deep Q Learning, howpublished = https://github.com/masfiq100/A-Reinforcement-Learning-Approach-for-Reducing-Traffic-Congestion-using-Deep-Q-Learning/tree/main/traffic/TLCS/intersection, year=,note=,.
67. Browne, C. B. et al. A survey of monte Carlo tree search methods. *IEEE Trans. Comput. Intell. AI Games* **4**, 1–43 (2012).

Acknowledgements
The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for paying the Article Processing Charges (APC) of this publication. This research is supported by Automated Systems and Soft Computing Lab (ASSCL), Prince Sultan University, Riyadh, Saudi Arabia. The authors would like to thank Prince Sultan University, Riyadh, Saudi Arabia for their support.

Author contributions
Conceptualization, S.M.R.S., S.N.N., P.M., V.P.M.; Methodology, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Software, S.M.R.S., S.N.N., P.M., V.P.M.; Validation, A.T.A., Z.H., M.T.; Formal analysis, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Investigation, A.T.A., Z.H., M.T.; Resources, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Data curation, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Writing-original draft, S.M.R.S., S.N.N., P.M., V.P.M.; Writing-review & editing, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Visualization, S.M.R.S., S.N.N., P.M., V.P.M., A.T.A., Z.H., M.T.; Supervision, V.P.M. and A.T.A.; Funding acquisition, M.T. All authors have read and agreed to the published version of the manuscript.

Funding
This research was funded by Prince Sultan University, Riyadh, Saudi Arabia.

Declarations

Competing interests
The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Additional information
Correspondence and requests for materials should be addressed to V.P.M., A.T.A. or M.T.